AF/243
 # IFW

THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT   :   Stefano Olivieri

SERIAL NO.   :   09/455,662   EXAMINER   :   Tung T. Vo

FILED   :   December 7, 1999   ART UNIT   :   2613

FOR   :   MOTION VECTOR ESTIMATION

APPEAL BRIEF TRANSMITTAL LETTER

Mail Stop Appeal Brief-Patents
Commissioner for Patents
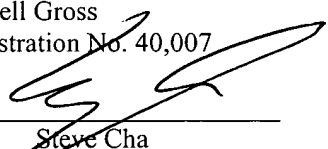P.O. Box 1450
Alexandria, VA. 22313-1450

Dear Sir:

Appellants respectfully submit three copies of a Brief For Appellants that includes an

Appendix with the pending claims. The Appeal Brief is now due on October 25, 2004.

Appellants enclose a check in the amount of $340.00 covering the requisite Government Fee.

Should the Examiner deem that there are any issues which may be best resolved by telephone

communication, kindly telephone Applicants undersigned representative at the number listed below.

Respectfully submitted,
Russell Gross
Registration No. 40,007

By: Steve Cha
Attorney for Applicant
Registration No. 44,069

Date: October 25, 2004

Mail all correspondence to:
Russell Gross, Registration No. 40,007
US PHILIPS CORPORATION
P.O. Box 3001
Briarcliff Manor, NY 10510-8001
Phone: (914) 333-9608
Fax: (914) 332-0615

Certificate of Mailing Under 37 CFR 1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to MAIL STOP APPEAL BRIEF-PATENTS, COMMISSIONER FOR PATENTS, P.O. BOX 1450, ALEXANDRIA, VA. 22313 on October 25, 2004.

Steve Cha, Reg. No. 44,069
(Name of Registered Rep.)   (Signature and Date)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Before the Board of Patent Appeals and Interferences

In re the Application

Inventor	:	Stefano Olivieri
Application No.	:	09/455,662
Filed	:	December 7, 1999
For	:	MOTION VECTOR ESTIMATION

APPEAL BRIEF

On Appeal from Group Art Unit 2613

Date: October 25, 2004

Russell Gross
Registration No. 40,007
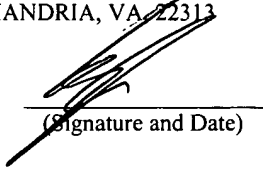By: Steve Cha
Attorney for Applicant
Registration No. 44,069

Certificate of Mailing Under 37 CFR 1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to MAIL STOP AMENDMENT COMMISSIONER FOR PATENTS, P.O. BOX 1450, ALEXANDRIA, VA 22313 on October 25, 2004.

Steve Cha, Reg. No. 44,069
(Name of Registered Rep.)

_____
(Signature and Date)

TABLE OF CONTENTS

I.   REAL PARTY IN INTEREST

The real party in interest is the assignee of the present application, Koninklijke Philips Electronics N.V., and not the party named in the above caption.

II.   RELATED APPEALS AND INTERFERENCES

With regard to identifying by number and filing date all other appeals or interferences known to Appellant which will directly effect or be directly affected by or have a bearing on the Board's decision in this appeal, Appellant is not aware of any such appeals or interferences.

III.   STATUS OF CLAIMS

Claims 1-3 have been presented for examination. All of these claims are pending, stand finally rejected, and form the subject matter of the present appeal.

IV.   STATUS OF AMENDMENTS

No amendment After Final Rejection has been filed.

V.   SUMMARY OF THE INVENTION

The present invention relates to generating motion vectors in the compression of video (page 1, lines 1-6). To send a video sequence of frames or "snapshots" from a transmitter to a receiver, without compression, requires a lot of bandwidth. Accordingly, the prevailing strategy is to compress the video, transmit the

3

compressed video, and have the video decompressed back to original form at the receiver.

It has been observed that one frame or "snapshot" in a video sequence is generally similar to the immediately next frame (page 1, lines 3-6). If, for example, some object in the scene is moving, the position of that object will differ from frame to frame, although the interframe difference will tend to remain fairly constant. Likewise, the rest of the scene stays identical or changes little from one frame to the immediately next frame. Thus, to further save transmission bandwidth, it is common to compress merely the image difference between the two frames, to compress a vector representative of a change in position of a block within the frame, and to send this compressed information instead. This differencing operation may continue for a number of frames until full frame compression is resumed. The vector to be compressed is known as a motion vector.

If there has been no motion for the particular block, the same image information will exist in the other frame at the same frame coordinates. Thus, such a motion vector will be zero, indicating no motion.

If there has been motion, this is detected by comparing the image information in the block to respective blocks in the other frame. Once a match is found, the difference in coordinates defines a motion vector. The determination of a motion vector is known as motion estimation.

An exhaustive search can be made for the matching block by, for example, shifting the searching block one pixel to the left, right, top or bottom for each matching operation, and picking the block and corresponding displacement with the closest match.

Computation-reducing techniques have evolved, however, that pre-select a

4

relatively small number of specifically-chosen candidate blocks and then choose the best-matching block from among those candidates (page 1, lines 15-22). With either exhaustive-search or computational-reducing techniques, once a motion vector has been determined for the block, the technique is repeated for the next block of the frame, until the entire frame has been completed.

One type of computation-reducing technique is recursive in that the motion estimation algorithm computes the motion vector for a block by exploiting motion information already determined for previous blocks (page 9, lines 8-10). An example of a recursive motion vector generation technique is 3-Dimensional Recursive Search (3D-RS) (page 3, lines 19-24).

In determining its relatively small set of candidate motion vectors, 3D-RS generates two candidate vectors based on the motion vector that had been selected for the previous block, i.e., the left neighbor (page 4, lines 10-11). However, for compression according to the H.263 protocol, 3D-RS performance falters (page 5, lines 1-2).

An embodiment of the present invention likewise generates two candidate vectors based on the motion vector that had been selected for the previous block, i.e., the left neighbor (page 4, equation 1 and line 11; page 6, equation 2 and line 11).

Unlike the prior art, however, the present invention incorporates a refinement within the recursion loop, the results of that refinement being available in the next iteration, i.e., for the next-processed block (page 7, lines 7-14).

The present invention may be implemented for recursive operation with an estimation circuit E, a refinement circuit REF and a motion field memory MEM (FIG. 1). Based on a number of inputs, including vectors from the memory MEM from which

5

candidate vectors are to be generated, the estimation circuit E selects one of the candidate vectors. The circuit E generates, from the selection, a vector $d^1(b,t)$. The refinement circuit REF generates a plurality of test vectors from the vector $d^1(b,t)$ (page 5, lines 16-19). The refinement may involve adding +1, 0 or -1 to each component of the vector $d^1(b,t)$ to generate 8 surrounding candidate or test vectors (page 9, lines 11-16). The refinement circuit REF selects one of the test vectors to generate an output vector $d^2(b,t)$ and sends the output vector to storage MEM (page 5, lines 19-23).

In another embodiment, the difference $d^2(b,t) - d^1(b,t)$ can be used to generate an additional candidate motion vector (page 8, lines 1-8), although this is considered an optional feature merely additive to the considerable performance improvement afforded by the instant invention.

VI.   ISSUE

Whether claims 1-3 are patentable under 35 U.S.C. 102(e) over U.S. Patent No. 6,108,039 to Linzer et al. ("Linzer").

VII.   GROUPING OF CLAIMS

Claims 1 and 3 stand or fall together. Claim 2 stands or falls separately.

VIII.   ARGUMENT

Present claim 1 recites, "A recursive motion vector estimation method.

The instant specification defines the word "recursive" (page 9, lines 8-10:

"The word recursive means that the motion estimation algorithm computes the motion

vector associated with a picture portion (for example a block) by exploiting motion

information already determined for previous blocks."

Linzer fails to disclose or suggest this feature of present claim 1.

Moreover, as the following analysis demonstrates, Linzer fails to disclose

or suggest other language of the present claim 1.

As a preliminary matter, it is unclear what the Office Action deems to be

the "current block" of the present claim 1, although item 2 suggests in its reference on

page 3 to step f) that the "current block" of the present claim 1 is a Linzer macroblock.

Also, referring to Linzer (FIG. 5; col. 16, lines 29-31; col. 9, line 34 to col.

10, line 66), item 2 of the final Office Action apparently deems the candidate motion

vectors used to obtain motion vectors MEO-MVTT and MEO-MVBT of Linzer to be the

"plurality of candidate vectors" of the present claim 1 (see, for example, the reference to

b) on page 3 of item 2 of the Office Action).

The Linzer motion estimator processor 60 produces the Linzer "candidate

vectors" (col. 10, lines 55-66; FIG. 3).

The Linzer motion estimator 24 is made up of the motion estimator

processor 60 and two inputs to the processor, i.e., the search window memory (WMEM)

56 and the target block memory (TMEM) 58 (FIG. 3). At any given time, the TMEM 58

holds a single 8x8 block of pixel data (col. 8, lines 66-67). A macroblock is a 2x2

configuration of these blocks, and is therefore 16 pixels by 16 pixels (col. 3, lines 15-17).

A search range or "search window" covers the pixels within a macroblock and those

extending a predetermined distance beyond the macroblock in all four directions (col. 2,

lines 63-66). The WMEM 56 holds search window data for the picture containing the

8x8 block (col. 8, lines 63-64). In particular, the WMEM 56 holds the search window

data for the 8x8 block currently occupying the TMEM 58.

The motion estimator processor 60 performs motion estimation for each

8x8 block of the macroblock separately (col. 8, line 66 – col. 9, line 1: "Illustratively,

TMEM 58 holds only a single 8x8 block of pixel data at a time and motion estimation is

separately performed for each (luminance) block of the macroblock" [underlining added

for emphasis]; FIG. 3).

"A motion vector candidate is then determined for the entire macroblock

based on the motion estimation results for each block of the macroblock" (col. 9, lines 1-

4).

There is no disclosure or suggestion in Linzer of determining a "motion

vector candidate" of a macroblock by using "motion information" generated for any other

macroblock, let alone a "previously-processed block."

According to the above, Linzer fails to disclose or suggest, "generating (E)

a plurality of candidate vectors" "based on motion information generated for a

previously-processed block." Accordingly, for at least this reason, Linzer fails to

anticipate present claim 1.

For the present claim 1 feature "motion information generated for the

previously-processed block," item 2 of the Office Action cites (col. 2, line 65 – col. 3,

line 9; col. 9, line 46 – col. 10, line 16).

The first passage (col. 2, line 65 – col. 3, line 9) describes, in a general

way, conventional block-matching motion estimation (col. 3, line 9: "motion

estimation"). The search (col. 2, line 65: "search") could be exhaustive or computation-

reducing. In the latter case, fewer than all macroblocks in the search range are subject to matching (col. 2, line 67: "matching"). Conventional motion estimation determines a motion vector for a block of a current frame, and then re-executes the procedure for the next block of the current frame, but does not use "motion information" generated for a previous block in making the determination.

Linzer fails to disclose or suggest determining a "motion vector candidate" of a macroblock by using "motion information" generated for any other macroblock, let alone a "previously-processed block."

The second passage cited by item 2 of the Office Action, i.e., col. 9, line 46 – col. 10, line 16, relates to Linzer motion estimation for the current macroblock (col. 16, lines 36-37: "the to-be-encoded frame macroblock"; line 53: "the macroblock"). In particular, the processing shown in FIG. 4, the first embodiment, or in FIG. 5, the second embodiment, relates to the processing for a single macroblock. That processing is repeated for each macroblock. However, there is no disclosure or suggestion that the processing of one macroblock uses "motion information" generated in processing another macroblock.

Linzer FIG. 5, for example, sets forth a flow of processing that applies to a single macroblock (col. 16, lines 27-37; col. 17, lines 1-2), and the processing is then repeated for a next macroblock. Linzer fails to disclose or suggest that "motion information generated" for one macroblock is used in "generating (E) a plurality of candidate vectors" in a next macroblock, wherein the "motion information" has been generated for a block residing "immediately to the left of said current block [i.e., "macroblock" in Linzer]."

Also, at least due to the latter feature of being "immediately to the left," a "previously-processed block" cannot be found in Linzer FIG. 5 by comparing one stage to another (e.g., stage ME2 to ME1). Nor has the Office Action attempted to suggest such an interpretation.

In addition, although the Office Action has not offered such an interpretation, it would not be proper to characterize as the "current block" of the present claim 1 an 8x8 block constituent, i.e., quadrant, of a Linzer macroblock rather than the macroblock itself. Firstly, the "previously-processed block" for the upper or lower left-hand quadrant must, according to the present claim 1, lie "immediately to the left of said current block." This implies that the "previously-processed block" resides in an entirely different Linzer macroblock. This further implies, as set forth above, that such a purported "current block" of Linzer would not be determined based on "motion information" of the "previously-processed block" of the recursive method of present claim 1. Therefore, under this hypothetical interpretation too, Linzer fails to disclose or suggest, "generating (E) a plurality of candidate vectors" "based on motion information generated for a previously-processed block."

For the present claim 1 language, "recursive motion vector estimation, item 2 of the Office Action apparently cites to claim 9 of Linzer. Linzer claim 9 recites, "The method of claim 8 further comprising . . . repeating steps (b)-(e) for each of plural macroblocks."

Steps (b)-(e) of Linzer claim 8, however, fail to disclose or suggest using the "motion information" generated for a "previously-processed block.". This quoted language explicitly appears in the present claim 1.

Moreover, and as set forth above, the instant specification defines the

word "recursive" (page 9, lines 8-10: "The word recursive means that the motion

estimation algorithm computes the motion vector associated with a picture portion (for

example a block) by exploiting motion information already determined for previous

blocks."

In particular, and as presented above, Linzer fails to disclose or suggest:

> A recursive motion vector estimation method, comprising the steps of:
> a) for a current block of a picture divided into a plurality of blocks, and
> based on motion information generated for the previously-processed block
> if any and if immediately to the left of said current block, the blocks being
> processed by said method in a predetermined order, generating (E) a
> plurality of candidate vectors . . .; and
> f) re-executing steps a) through f) for a next-to-be-processed block, if any,
> as said current block.

For at least all of the above reasons, Linzer fails to anticipate the invention

as recited in claim 1. Nor, due to the very differing design of Linzer as compared to the

present invention, can the present applicants see how it fairly could be said that the

present invention as recited claim 1 would have been obvious over Linzer.

In addition, claim 1 recites, "A recursive motion vector estimation

method, comprising the steps of: a) generating (E) a plurality of candidate vectors from

stored vectors (PV); . . .; e) storing (MEM) the output vector (d^2); and f) re-executing

steps . . ."

As mentioned above, the instant specification defines the word "recursive"

(page 9, lines 8-10: "The word recursive means that the motion estimation algorithm

computes the motion vector associated with a picture portion (for example a block) by

exploiting motion information already determined for previous blocks."

11

The "recursive" nature of the present invention as recited in claim 1 implies that the generating of vectors in step a) is from "output" vectors stored in step e).

Linzer, by contrast, does not carry over generated motion vectors from the processing of one block to the processing of a next block. Linzer, therefore, does not disclose the recursive "storing" of the output vector and "generating" from "stored vectors" block-to-block of the present invention as recited in claim 1.

Item 2 of the Office Action deems the Linzer motion vectors supplied by motion estimator 24 (col. 8, line 14-16; FIG. 2) to be the "stored vectors" of present claim 1.

However, with regard to the present claim 1, Linzer fails to disclose or suggest the "recursive" "storing" of the output vector in step e), followed by the "generating" in step a), in accordance with the explicit definition of the term "recursive" in the specification.

Claim 2 likewise recites, "A recursive motion vector estimation method, comprising the steps of: a) generating (E) a plurality of candidate vectors from stored vectors (PV); . . .; e) storing (MEM) the output vector (d^2) . . ."

Although claim 2 does not explicitly recite the repetition step f) of claim 1, this repetition is implicit in the definition of the term recursive, defined in the specification, and in claim 2 as a whole.

As set forth above, the instant specification defines the word "recursive" (page 9, lines 8-10: "The word recursive means that the motion estimation algorithm computes the motion vector associated with a picture portion (for example a block) by exploiting motion information already determined for previous blocks."

Additionally, the step e) of storing the "output" vector is the last step in the "recursive" method, a plurality of candidate vectors being generated from "stored vectors." Linzer shows no corresponding "output" vector under any reasonable interpretation.

The Office Action suggests that, Linzer processing from the top of FIG. 5 to the bottom, i.e., of a single macroblock, is one iteration in a Linzer "recursive" method, but this interpretation attempts to ignore the explicit definition of the "recursive" provided by the applicant in the instant specification.

For at least the above reasons, present claim 2 distinguishes patentably over Linzer without having to recite that the previously processed block resides immediately to the left, as is recited in present claim 1.
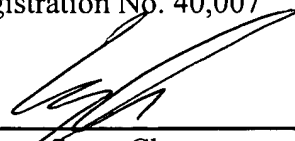
Claim 3 is an apparatus claim based on method claim 1, and is likewise deemed to be patentable.

IX. CONCLUSION

In view of the above analysis, it is respectfully submitted that the referenced teachings, whether taken individually or in combination, fail to anticipate or render obvious the subject matter of any of the present claims. Therefore, reversal of all outstanding grounds of rejection is respectfully solicited.

Respectfully submitted,

Russell Gross
Registration No. 40,007

Date: October 25, 2004

By: Steve Cha
Attorney for Applicant
Registration No. 44,069

X. APPENDIX: THE CLAIMS ON APPEAL

1. A recursive motion vector estimation method, comprising the steps of:

a) for a current block of a picture divided into a plurality of blocks, and based on motion information generated for the previously-processed block if any and if immediately to the left of said current block, the blocks being processed by said method in a predetermined order, generating (E) a plurality of candidate vectors from stored vectors (PV);

b) selecting (E) one of these candidate vectors to generate a selected vector (d^1);

c) generating (REF) a plurality of test vectors from the selected vector (d^1);

d) selecting (REF) one of the test vectors to generate an output vector (d^2);

e) storing (MEM) the output vector (d^2); and

f) re-executing steps a) through f) for a next-to-be-processed block, if any, as said current block.

2. A recursive motion vector estimation method, comprising the steps of:

generating (E) a plurality of candidate vectors from stored vectors (PV);

selecting (E) one of these candidate vectors to generate a selected vector (d^1);

generating (REF) a plurality of test vectors from the selected vector (d^1);

selecting (REF) one of the test vectors to generate an output vector (d^2); and

storing (MEM) the output vector (d^2), wherein said step of generating a plurality of test vectors from the selected vector (d^1) includes the step of adding -1, 0, or +1 to each component of the selected vector (d^1).

3. A device for recursive motion vector estimation, the device comprising:

a) for a current block of a picture divided into a plurality of blocks, and based on motion information generated for the previously-processed block if any and if

immediately to the left of said current block, the blocks being processed by said method in a predetermined order, means (E) for generating a plurality of candidate vectors from stored vectors;

b) means (E) for selecting one of these candidate vectors to generate a selected vector (d^1);

c) means (REF) for generating a plurality of test vectors from the selected vector (d^1);

d) means (REF) for selecting one of the test vectors to generate an output vector (d^2);

e) means (MEM) for storing the output vector (d^2); and

f) means for re-executing steps a) through f) for a next-to-be-processed block, if any, as said current block.